Package: CNVMotif (via r-universe)

July 2, 2024

Title Explore, Analyze and Visualize Catalogs and Patterns of Copy Number Variation in Cancer Genomics

Version 0.1.0

Date 2020-07-16

Description Provides functionality for exploring, analyzing and visualizing the copy number variation (CNV) motifs in cancer genomics.

License MIT + file LICENSE

Imports Biostrings, cluster, data.table, dplyr, factoextra, ggplot2, ggseqlogo, magrittr, msa, purrr, Rcpp, tidyr

Suggests covr, doParallel, foreach, ggmsa, ggpubr, scales, testthat

LinkingTo Rcpp

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE, roclets = c(``collate", ``namespace", ``rd", ``roxytest::testthat_roclet"))

RoxygenNote 7.1.1

Repository https://shixiangwang.r-universe.dev

RemoteUrl https://github.com/ShixiangWang/CNVMotif

RemoteRef HEAD

RemoteSha ad11585b11b89d7970552f9a805c0b5db9d669c3

Contents

`build_sub_matrix` *Build a Substitution Matrix*

Description

Build a Substitution Matrix

Usage

```
build_sub_matrix(simple_version = FALSE, max_len_score = 4L)
```

Arguments

`simple_version`	if `TRUE`, just use segmental copy number value.
`max_len_score`	the maximum score for segment length (should >=4). The maximum score for copy number value is 6 (cannot be changed).

Value

a `list`.

Examples

```
sub_list <- build_sub_matrix()
sub_list2 <- build_sub_matrix(simple_version = TRUE)
```

`cluster_pam_estimate` *Estimate Optimal Number of Cluster for PAM Algorithm*

Description

`cluster::clusGap()` cannot be used here for distance matrix, so it is removed.

Usage

```
cluster_pam_estimate(
  x,
  method = c("silhouette", "wss"),
  k.max = 10,
  verbose = interactive(),
  barfill = "steelblue",
  barcolor = "steelblue",
  linecolor = "steelblue",
  FUNcluster = cluster::pam,
  seed = 1234L,
  clean_memory = FALSE,
  ...
)

cluster_pam(x, k, ...)
```

Arguments

x	a dissimilarity matrix.
method	the method to be used for estimating the optimal number of clusters. Possible values are "silhouette" (for average silhouette width), "wss" (for total within sum of square) and "gap_stat" (for gap statistics).
k.max	the maximum number of clusters to consider, must be at least two.
verbose	logical value. If TRUE, the result of progress is printed.
barfill	fill color and outline color for bars
barcolor	fill color and outline color for bars
linecolor	color for lines
FUNcluster	a partitioning function which accepts as first argument a (data) matrix like x, second argument, say k, k >= 2, the number of clusters desired, and returns a list with a component named cluster which contains the grouping of observations. Allowed values include: kmeans, cluster::pam, cluster::clara, cluster::fanny, hcut, etc. This argument is not required when x is an output of the function `NbClust::NbClust()`.
seed	random seed.
clean_memory	logical. If `TRUE`, the cluster result object will be removed and the memory will be released by calling `gc()` to reduce the memory consumption.
...	other parameters passing to cluster::pam.
k	positive integer specifying the number of clusters, less than the number of observations.

Value

a `ggplot` object.

a PAM clustering result object.

Examples

```
data("iris")
head(iris)
iris.scaled <- scale(iris[, -5])
iris.dist <- dist(iris.scaled) %>% as.matrix()
p <- cluster_pam_estimate(iris.dist)
p2 <- cluster_pam_estimate(iris.dist, method = "wss")

cl <- cluster_pam(iris.dist, 3)
```

---

cluster_split *Split Cluster Sequence into List*

Description

Split Cluster Sequence into List

Usage

```
cluster_split(x, s = NULL, block_size = 10)
```

Arguments

x	a named integer vector from hclust etc.
s	default is NULL, when the x is block cluster sequence, set this to a sequence vector.
block_size	block size used to split, only used when s is not NULL.

Value

a list.

---

do_msa *Run Modified Multiple Sequence Alignment*

Description

Run Modified Multiple Sequence Alignment

Usage

```
do_msa(
  x,
  substitutionMatrix = NULL,
  gapOpening = 6,
  gapExtension = 1,
  verbose = FALSE,
  ...
)
```

Arguments

x	a character vector.
substitutionMatrix	substitution matrix for scoring matches and mismatches. Default is NULL, use 1 for match and 0 for unmatch.
gapOpening	gap opening penalty; Note that the sign of this parameter is ignored.
gapExtension	gap extension penalty; Note that the sign of this parameter is ignored.
verbose	if TRUE, print extra info.
...	other arguments passing to msa::msa

Value

a list.

Examples

```
r <- do_msa(c("ABCDF", "BCDEF"))
r
```

---

extract_seqs *Extract Pasted Sequences from Each Chromosome*

Description

See get_score_matrix() for examples. The result sequences are unique and sorted.

Usage

```
extract_seqs(
  dt,
  len = 5L,
  step = 1L,
  local_cutoff = 1e+07,
  flexible_approach = FALSE,
  return_dt = FALSE
)
```

Arguments

dt	a data.table from transform_seqs.
len	cut length.
step	step size to move on each chromosome sequence.
local_cutoff	any segment with length greater than this cutoff will be filtered out and used as cutpoint, default is 10Mb.
flexible_approach	if TRUE, extract flexible-size sequences between segments with size less than specified cutoff. So the arguments len and step are ignored.
return_dt	if TRUE, just return a data.table containing mutated Seqs column.

Value

a list.

---

get_score_matrix *Get Copy Number Sequence Similarity or Distance Matrix*

Description

Get Copy Number Sequence Similarity or Distance Matrix

Usage

```
get_score_matrix(
  x,
  sub_mat = NULL,
  simple_version = FALSE,
  block_size = NULL,
  dislike = FALSE,
  cores = 1L,
  verbose = FALSE
)
```

Arguments

x	a coding copy number sequence (valid letters are A to X).
sub_mat	default is NULL, use longest common substring method. It can be a substitution matrix, each element indicates a score to plus. See build_sub_matrix().
simple_version	if TRUE, just use segmental copy number value.
block_size	a block size to aggregrate, this is designed for big data, it means results from adjacent sequences will be aggregrate by means to reduce the size of result matrix.
dislike	if TRUE, returns a dissimilarity matrix instead of a similarity matrix.
cores	computer cores, default is 1, note it is super fast already, set more cores typically do not speed up the computation.
verbose	if TRUE, print extra message, note it will slower the computation.

Value

a score matrix.

Examples

```
load(system.file("extdata", "toy_segTab.RData",
  package = "CNVMotif", mustWork = TRUE
))
x <- transform_seqs(segTabs)
x
seqs <- extract_seqs(x$dt)
seqs
seqs2 <- extract_seqs(x$dt, flexible_approach = TRUE)
seqs2

mat <- get_score_matrix(seqs$keep, x$mat, verbose = TRUE)
mat

mat2 <- get_score_matrix(seqs$keep, x$mat, dislike = TRUE)
identical(mat2, 120L - mat)

mat_b <- get_score_matrix(seqs$keep, x$mat, block_size = 2L)
## block1 represents the first 2 sequences
## block2 represents the 3rd, 4th sequences
## ...
mat_b

mat_c <- get_score_matrix(seqs$keep)
mat_c
mat_d <- get_score_matrix(seqs$keep, dislike = TRUE)
mat_d

if (requireNamespace("doParallel")) {
  mock_seqs <- sapply(1:10000, function(x) {
    paste(sample(LETTERS[1:24], 5, replace = TRUE), collapse = "")
  })

  system.time(
    y1 <- get_score_matrix(mock_seqs, x$mat, cores = 1)
  )

  system.time(
    y2 <- get_score_matrix(mock_seqs, x$mat, cores = 2)
  )

  all.equal(y1, y2)
}
```

ggseqlogo2 *Quick sequence logo plot*

Description

ggseqlogo is a shortcut for generating sequence logos. It adds the ggseqlogo theme theme_logo by default, and facets when multiple input data are provided. It serves as a convenient wrapper, so to customise logos beyond the defaults here, please use geom_logo.

Usage

```
ggseqlogo2(
  data,
  facet = "wrap",
  scales = "free_x",
  ncol = NULL,
  nrow = NULL,
  idor = NULL,
  ...
)

geom_logo2(
  data = NULL,
  method = "bits",
  seq_type = "auto",
  namespace = NULL,
  font = "roboto_medium",
  stack_width = 0.95,
  rev_stack_order = F,
  col_scheme = "auto",
  low_col = "black",
  high_col = "yellow",
  na_col = "grey20",
  plot = TRUE,
  idor = NULL,
  ...
)
```

Arguments

data	Character vector of sequences or named list of sequences. All sequences must have same width
facet	Facet type, can be 'wrap' or 'grid'
scales	Facet scales, see facet_wrap
ncol	Number of columns, works only when facet='wrap', see facet_wrap
nrow	Number of rows, same as ncol

idor	a named vector (like a dictory) to change letters one to one in the plot.
...	Additional arguments passed to geom_logo
method	Height method, can be one of "bits" or "probability" (default: "bits")
seq_type	Sequence type, can be one of "auto", "aa", "dna", "rna" or "other" (default: "auto", sequence type is automatically guessed)
namespace	Character vector of single letters to be used for custom namespaces. Can be alphanumeric, including Greek characters.
font	Name of font. See list_fonts for available fonts.
stack_width	Width of letter stack between 0 and 1 (default: 0.95)
rev_stack_order	If TRUE, order of letter stack is reversed (default: FALSE)
col_scheme	Color scheme applied to the sequence logo. See list_col_schemes for available fonts. (default: "auto", color scheme is automatically picked based on seq_type). One can also pass custom color scheme objects created with the make_col_scheme function
low_col	Colors for low and high ends of the gradient if a quantitative color scheme is used (default: "black" and "yellow").
high_col	Colors for low and high ends of the gradient if a quantitative color scheme is used (default: "black" and "yellow").
na_col	Color for letters missing in color scheme (default: "grey20")
plot	If FALSE, plotting data is returned

Examples

```
library(ggseqlogo)
data(ggseqlogo_sample)

## Same as ggseqlogo()
p1 <- ggseqlogo2(seqs_dna[[1]])
p1

## Extra feature
idor <- as.character(1:4)
names(idor) <- c("A", "C", "G", "T")
p2 <- ggseqlogo2(seqs_dna[[1]], idor = idor)
p2
```

---

show_seq_logo *Show Copy Number Sequence Logos*

Description

Show Copy Number Sequence Logos

Usage

```
show_seq_logo(
  x,
  method = c("prob", "bits"),
  simple_version = FALSE,
  ncol = NULL,
  nrow = NULL,
  recode = FALSE,
  indicator = NULL,
  ...
)
```

Arguments

x	a character vector of sequences or named list of sequences. All sequences must have same width.
method	Height method, can be one of "bits" or "probability" (default: "bits")
simple_version	if TRUE, just use segmental copy number value.
ncol	Number of columns, works only when facet='wrap', see facet_wrap
nrow	Number of rows, same as ncol
recode	if TRUE, it will use default indicator or specified indicator to show the letters in the plot
indicator	a named vector (like a dictory) to change letters one to one in the plot.
...	Additional arguments passed to geom_logo

Value

a ggplot object

Examples

```
p1 <- show_seq_logo(sapply(split(LETTERS[1:24], 1:4), function(x) paste0(x, collapse = "")))
p1
p2 <- show_seq_logo(sapply(split(LETTERS[1:24], 1:4), function(x) paste0(x, collapse = "")),
  recode = TRUE
)
p2
p3 <- show_seq_logo(sapply(split(LETTERS[1:6], 1:2), function(x) paste0(x, collapse = "")),
  simple_version = TRUE
)
```

show_seq_shape *Show Copy Number Sequence Shapes*

Description

Show Copy Number Sequence Shapes

Usage

```
show_seq_shape(
  x,
  map = NULL,
  simple_version = FALSE,
  line_size_scale = 3,
  x_lab = ifelse(simple_version, "Assumed equal length", "Estimated segment length"),
  y_lab = "Copy number",
  nrow = NULL,
  ncol = NULL,
  scales = "free_x"
)
```

Arguments

x	a character vector of sequences or named list of sequences. All sequences must have same width.
map	default is NULL, a named string vector.
simple_version	if TRUE, just use segmental copy number value.
line_size_scale	the scale size for line width.
x_lab	x lab.
y_lab	y lab.
nrow	Number of rows, same as ncol
ncol	Number of columns, works only when facet='wrap', see facet_wrap
scales	Should scales be fixed ("fixed", the default), free ("free"), or free in one dimension ("free_x", "free_y")?

Value

a ggplot object.

Examples

```
p <- show_seq_shape(c("ADGHK"))
p

x <- list(a = c("ABCDE", "AXFDP"), b = c("KKDFH", "GKDFM"))
p2 <- show_seq_shape(x)
p2

p3 <- show_seq_shape(c("ABCD"), simple_version = TRUE)
p3
```

`transform_seqs` *Coding Copy Number Segments with Letters.*

Description

See `get_score_matrix()` for examples. See details for full description of implementation.

Usage

```
transform_seqs(x, simple_version = FALSE, max_len_score = 4L)
```

Arguments

`x`	a `CopyNumber` object or a `data.frame` with at least 5 columns ("sample", "chromosome", "start", "end", "segVal").
`simple_version`	if `TRUE`, just use segmental copy number value.
`max_len_score`	the maximum score for segment length (should >=4). The maximum score for copy number value is 6 (cannot be changed).

Details

For complicated cases, letters are grouped as short (<50kb), mid (<500kb), long (<5Mb), long (or extreme) long (>5Mb) segments.

- A B C D for copy number 0.
- E F G H for copy number 1.
- I J K L for copy number 2.
- M N O P for copy number 3.
- Q R S T for copy number 4.
- U V W X for copy number 5+.

For simplified cases, letters are used to code only segment copy number value.

- A for copy number 0.
- B for copy number 1.

- C for copy number 2.
- D for copy number 3.
- E for copy number 4.
- F for copy number 5+.

Value

a `list`.

Index